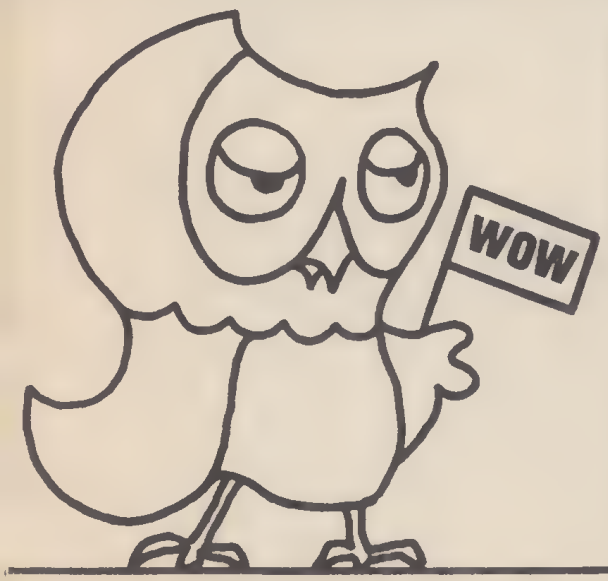
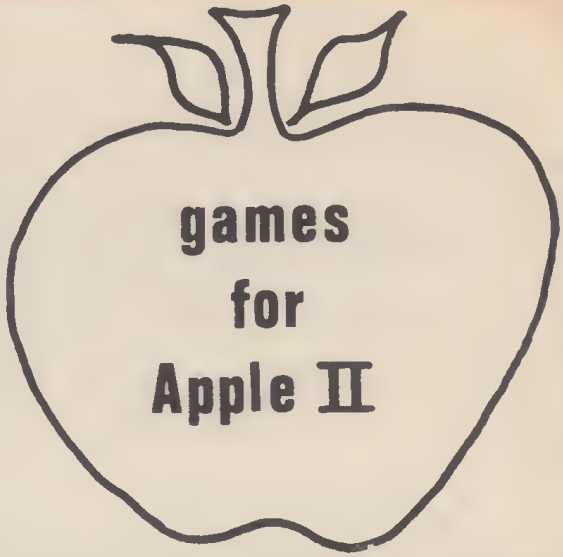
SIDE 1

WORDSMITH

word game

16 K





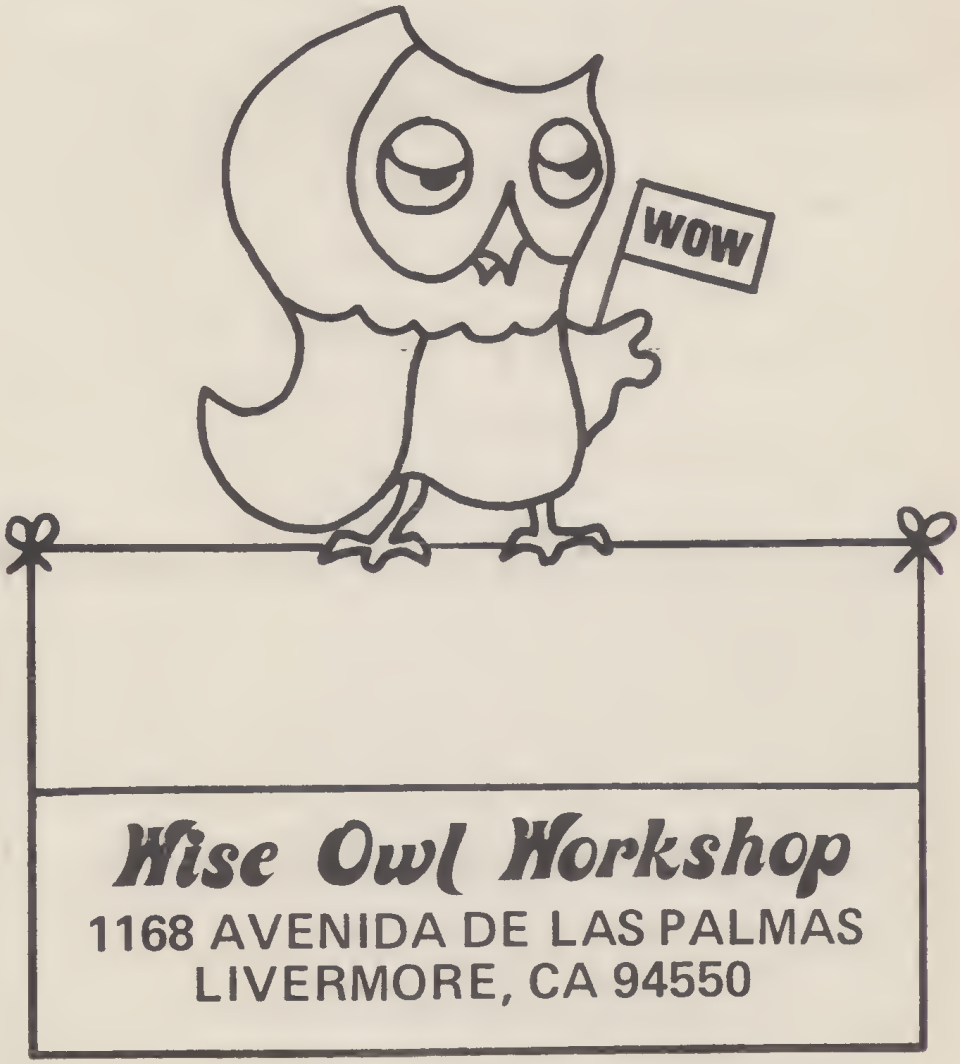
SIDE 2

HUNT THE WUMPUS

classic computer game

16 K

COMPLETE WITH CUSTOMIZING INSTRUCTIONS
AND PROGRAMMING HINTS


WOW
Wise Owl Workshop
1168 AVENIDA DE LAS PALMAS
LIVERMORE, CA 94550

WORDSMITH

A one or two person game where the object is to guess the word selected by the other player or the computer. The scoring makes exciting, come-from-behind victories possible.

SETUP

A. Load the program with a >LOAD command.
B. Start the program with a >RUN command.
C. Select the one or two person game.
D. Enter the names of the players.
E. Five words must be entered, in secret, for each player. You can press RETURN, (only) and get one of the 475 words stored in the program. This is the normal way to get the words for the solitaire game. The program will not select the same word more than once as long as the program is not restarted with a RUN command.

PLAY

A. After each side has selected 5 words, the game will begin.
B. The computer will display each side of the game and 10 dots to represent the word to be guessed.
C. The game is played by guessing a letter or letters that might be in the word.
D. Each turn costs 10 points.
E. For a single letter guess, the point values of all of the letters are displayed beside the letter in the value table. For example: the letter Y has a value of 40 points. Guess a letter or a blank, and if that letter is in the word and not already identified, the first or last occurance of that letter will be displayed and the points added to the score.

F. For multiple letter guesses, the points are computed from the point value of all of the unidentified letters multiplied by the number of unidentified letters in the guess. For example: assume the word is BRAKE and by single letter guess you have determined the following.

..BR..E...

The entry would be:AK
The scoring for that play is:

```
  A = 10
+ K = 50
-------
   60
 *  2(number of letters)
-------
  120 points
```

G. Even longer guesses can be made by use of the semi-colon (;) to substitute for letters already identified. In the above example the following entry could be made:ØØ;;AK;ØØØ

The semi-colons stand for the already identified letters BR and E. (The Ø symbols indicate blanks the actual entry would appear as: ;;AK;)

The scoring would be:

```
Ø =  10
Ø =  10
B =   0 (already identified)
R =   0 (already identified)
A =  10
K =  50
E =   0 (already identified)
Ø =  10
Ø =  10
Ø =  10
-------
 110
 *   7 (unidentified letters)
-------
 770 points
```

H. If a letter is identified, it is turned to flashing mode in the value table. If a letter has been found not to appear any more in the word, it is erased from the value table.

I. Each player continues playing until he makes an incorrect guess, then it is the other players turn.

J. The game continues until one side gets all of the blanks and letters in his word. The other player gets two additional turns to guess his word and the game ends. (The right side player gets 3 guesses after the left side player finishes, since he has not yet finished the current turn.)

K. A challenge is allowed for each word where the players decide if both words were legitimate. If a word is not legitimate, the player entering the bad word cannot make any points that round and his opponent connot lose any points.

L. At the end of 5 rounds, each player is given a title from Dunce (worst) to Wordsmith (best).

VARIATIONS

The normal game is played using common words with no proper words allowed. Other variations can be used to make more interesting games or for educational purposes. The following variations might be tried.

A. Cities of the World	D. Actors
B. Cities of the U.S.	E. Automobiles
C. Famous Scientists	F. Politicians

Since the program only handles up to 10 character words, only the first 10 characters of the name could be used.
For example: SAN FRANCISCO would be SAN FRANCI

CUSTOM MODIFICATIONS

WORDS

You can insert your own word set to replace the stored set. The words are stored by the statements from 8300 to 8481.

SCORING

The scoring in the value table can be changed in statements from 8500 to 8530 and the value of the blanks in statements 580 and 675.

PROGRAMMING HINTS

Programs occasionally need to select items from a list, once and only once in a program. For example: The words in WORDSMITH are not used more than once unless the program is restarted. The method that uses the least amount of storage and is simple is called destructive removal. This is the method used in WORDSMITH.

When a word is randomly selected from the list and transferred to another array, it is destroyed. In WORDSMITH this is accomplished by changing the second letter of the word to the @ sign. If that word is selected randomly again, the second letter of the word is checked for the @ sign. If it appears, another word is chosen until a good word (no @ sign is found). The storage and check for the @ sign is found in statement 430.

The disadvantage of this method is speed of execution since the program will spend a lot of time looking for a good word when very few good words are left.

HUNT THE WUMPUS GAME

Hunt the horrible Wumpus as he sleeps in a maze of caves. Hazards abound but there is a chance to kill the Wumpus before he wakes up and eats you.

SETUP

A. Load the program with a >LOAD command.

B. Start the program with a >RUN command.

C. Enter your name.

D. The game will be set up with a maze of 20 rooms in a cave. There is a Pit, a Wumpus, and a Giant Bat in the cave. Each one starts out in a different room from the other hazards and from where you are placed.

E. You can hear the hazards if they are in a room next to you.

F. Each room has listed the connections to adjacent rooms.

PLAY

A. There are four different actions you can take in each room.

1. W=WALK. Walk into an adjoining room. You must enter the number of the room you want to enter.
2. T=THROW. Throw a rock into the next room. You must enter the number of the room you want to throw into. If the Wumpus is in the room that you throw into, he will wake up and enter another room, maybe the pit, maybe yours.

PROGRAMMING HINTS

SOUNDS

Sounds add a lot to programs, particularly one such as Wumpus. The sounds in Wumpus are generated by the following statements.

7400 Wumpus Snoring Sound
7910 Bat Squeeks
7920 Cricket Chirping
7930 Falling Sound
7940 Rising Sound
8300 Proclamation Bugle Call
Call 768 Growling Noise

You can see from this variety that a number of sounds can be generated by calls to the tone generator from BASIC. The sound generators are poked into memory by the statements from 9990 to 9995. The snoring sound is generated by poking the speaker location from BASIC.

If any problems are encountered with these programs or you have any questions, write to:

WISE OWL WORKSHOP
1168 Avenida De Las Palmas
Livermore, CA 94550